JACG

NEWSLETTER
Vol.2 No. 4
Dec. 1982
Single Copy
Price $1.00

THE JERSEY ATARI COMPUTER GROUP

From the Editor's Desk....

Well, here it is - December - already! It is difficult for me to believe that I have been your editor for four months. Back in July, when I tentatively volunteered to be the editor of the JACG Newsletter, I had no conception of either how to do it or the amount of time it would take. Now, I know. The last few months have taught me many things about what it takes to put together a deceptively small (it's only 12 pages) newsletter - editing, paste-up, printing, graphic arts, and TIME. The list could go on but the largest component of the entire process is time. I spend 15-20 hours a month on the newsletter, not counting time I spend at the photostat house, at the printers and talking to other club members and dealers.

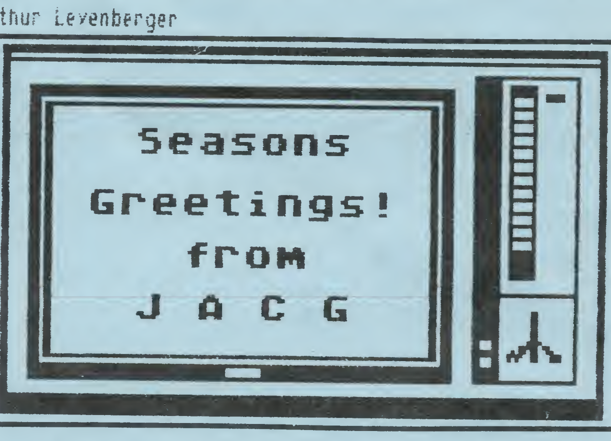
It does require a great amount of time but I believe it is worth it. Personal computers, microcomputers, home computers, whatever you want to call them, have come a long way since I first became interested in them during the mid-seventies. The technology and the hardware has advanced but the user is essentially still in the same position that s/he was 5 years ago. That is why we all belong to a user group like JACG. The best (and sometimes only) source of information concerning the operation and use of a computer like the Atari or Apple or IBM is from other users. This is why I feel that it is worth the effort to publish a newsletter that contains information that will be of help to all users.

My personal goal in having an Atari computer is to become a knowledgeable and efficient USER. I don't want to be a super programmer (I've been there and know what that entails). I also do not want to become a "hardware type" and be able to break the worlds speed record at disassembling and re-assembling my computer while blindfolded. No, those extremes are not for me. I do want to do a little word processing, a little database work, some graphics things, learn more about computing and maybe play a few games on the side.

These are hectic times. Not just for me attempting to get this paper out on time every month but for all of us Atari users. I thought the movie, Star Wars, was great but compared to the computer wars that are going on right now the movie lacks action and a substantial plot. Commodore comes out with a new machine, Atari lowers its prices. TI lowers its prices, Atari offers rebates. TI offers rebates (they can't give it away), Atari starts up the rumor mill about a "New Machine" being announced in January. This is the stuff of great fiction. I don't know where it will end but what I do know is that it will be 10 years before I have mastered the capabilities of my Atari 800 computer, regardless of what Madison Avenue Techno-junkies try to tell me.

As you can see, this issue of the newsletter is substantially larger than normal. This is our special 20-page Christmas issue which will be approximately 22 to 26 pages in length. I don't know the exact page count because I am writing this about a week before I do the paste-up. In any case, I felt that it was appropriate to get caught up with the material I have in my backlog by "going all the way" with an end-of-year, megapage, JACG Newsletter. I hope you like it, and to all, a good byte.

Arthur Levenberger



TWAS THE NIGHT BEFORE CHRISTMAS AND ALL THROUGH THE PAD
NOT A HIP CAT WAS SWINGING, AND THATS NOWHERE DAD,
THE STOVE WAS HUNG UP IN THAT STOCKING ROUTINE,
IN HOPES THAT THE FAT MAN WOULD SOON MAKE THE SCENE.
THE KIDS HAD ALL HAD IT SO THEY HIT THEIR SACKS,
AND ME AND THE BRIDE HAD BEGUN TO RELAX,
WHEN THERE STARTED A RUMBLE THAT CAME ON REAL FRANTIC
SO I OPENED THE WINDOW TO FIGURE THE PANIC.
I SAW A SQUARE SHORT THAT WAS MAKIN FAT TRACKS,
BEIN PULLED BY EIGHT DOGS WHO WAS WEARING HAT RACKS,
AND A FUNNY OLD GEEZER WAS FLIPPIN HIS LID,
HE TOLD EM TO MAKE IT, AND MAN, LIKE THEY DID.
I COULDNT HELP DIGGIN THE SCENE ON THE ROOF
AS I STOOD THERE JUST WAITING FOR CHUBBY TO GOOF,
THEY STOOD BY THE CHIMNEY IN BUNCHES AND CLUSTERS
TILL TUBBY SLID DOWN, COMIN ON LIKE GANGBUSTERS.
HIS THREADS WERE THE SQUAREST AND I HAD TO CHUCKLE,
IN FRONT NOT IN BACK WAS HIS IVY LEAGUE BUCKLE,
THE MOP ON HIS CHIN HID HIS BUTTON-DOWN COLLAR
AND WITH THAT RED NOES, MAN HE LOOKED LIKE A BALLER.
LIKE HE WAS THE SQUAREST, THE MOST ABSOLUTE,
BUT FACE IT, WHO CARES WHEN HE LEFT ALL THAT LOOT.
HE LAID THE JAZZ ON ME AND FLED FROM THE GIG,
WAILIN HAVE A COOL YULE AND MAN, LATER, LIKE DIG.

AND NOW A WORD FROM THE PRESIDENT.....

We keep getting bigger and better! Our November meeting, attended by nearly 300 enthused Atarians, was our best yet. We held our elections for the September 1982 to September 1983 year and your new slate of officers will be the same as your old slate of officers with the addition of Richard Rospond as Program Chairman. Our tape library should begin to really roll now that we have our tape duplication machine in hand, so all you tape users should be getting access to some of our terrific library stuff by next meeting. Our Editor, Art Leyenberger, announced that our December newsletter- which you now have in your hot little hands, will be much expanded to celebrate the holiday season. Not a bad piece of work, don't you think!!!

We got a look at the SCOPY 810 program that JACG is selling to its member and 18 copies (all we had at the time) were sold in minutes (this is a real bargain at $6 for the disk with extensive documentation). We also added three new game disks to our library thanks to Manny Lieberman's work and also added a new, improved FORTH disk and these, too, were sold out. Yours truly lamented that On-Line had made a mistake and sent our Frogger order AGAIN, and when I mentioned that it would be nice if I could sell them and not have to return them, I was mobbed and these, too, were gone in seconds.

We then got to the meat of the program. We began with a showing of the Atari video tape #1, featuring Chris Crawford as a 6502 microprocessor. We had shown this tape about nine months ago (when we were about 40 strong) and I decided that it would be nice for all our new members to see it, too. It was even fun to watch the second time around and we look forward to getting #2 and #3 soon.

Don Ursem, our chief librarian, then gave a terrific talk entitled "All You Ever Wanted To Know About Joysticks, But Were Afraid To Ask." He not only showed many types of joysticks, he also described how the various types work. He then proceeded to sell off some of them!! (Clever entrepreneur, that Don Ursem.)

Nick Scalera then gave very through and informative demos of Centipede, 3R Math Tutor and Salmon Run. This was followed by demos of Stellar Shuttle, Hellcat Ace and Chopper Rescue by Art Leyenberger. Once again, however, time got the better of us and we missed Art's long awaited and much prepared demo of Graphics Master. Also Fernando Herrera was not able to provide Astro Chase. The good news is that this gives us a start on the next meeting!

There is no doubt that the demos we have at meetings are helpful to our members in making purchasing decisions. I look forward to more demos by more members in the future. Contact our program Chairman, Richard Rospond, if you would like to demo something or give a talk or participate at meetings in any manner.

Because this newsletter will be delivered to you before the December meeting, let me give you an advanced preview of it. We will have a talk on FORTH by Jim Stanard. I think you will be intrigued by the capabilities of this language and, hopefully, less confused by its implementation. We also hope to have a talk by a member of the Atari advanced research group from NYC. (Yes, that is correct, Atari is not glued to the west coast.) Demos will, of course, also be much in evidence.

Please note two changes in meeting format that will take place beginning with this December meeting. Commercial sales of computer related items will be restricted to before the meetings (9:15-10:00) and there will be a general question and answer session from 9:30-10:00.

We took a ten minute break in the middle of th November meeting so that those who were interested could get library disks and whatever else was for sale and it turned out that several members had used software that they also sold. We will continue this practice in the future (as long as it does not get out of hand). If you have any used software that you no longer need (ORIGINALS ONLY!), bring it along and see if you can find a buyer.

Our secretary, Ed Picciuti, who is another budding entrepreneur, provided me with a modification for Epson printers that lifts it off a table so that your paper supply can go underneath and tilts it so that the printing is more visable without craning your neck. He is selling this clever item, so contact him if you are interested.

Atari's user support group will not be visiting us this fall due to lack of travel money, but they plan on making two visits to the east coast next year, one of which will include a visit to JACG. Also rumors regarding the "Atari 1000" are rampant - words like "January", "64K", "CP/M" and "800 compatible" are being bandied about. In the meantime, there is a virtual software blizzard for the 800. I went through a couple of recent magazines, and compiled a two page list of programs I have never seen or heard of that are being advertised. This is significant because, as president, I get a lot of literature and I try to keep up with the latest stuff for our computer. Even so, I could make a list that long! Now maybe you can appreciate the tremendous value of demos at our meetings.

A personal request to all other user groups (as well as our own members) - is anyone as annoyed as I am by the placement of the "atari" key on the keyboard and how it messes up reaching for the shift key when you touch type? Has anyone tried to modify the keyboard to switch the atari and shift keys? H anyone read this column and gotten to this poi Let me know!

Enough said for this month- Happy Holidays to all and may next year be a good one for all of you.

Dick Kushner-JACG President

WICO Command Controller
Reviewed by Ed Picciuti - JACG

The Wico joystick controller, which retails for $27.95, is an Atari compatible joystick controller with a bat style stick and 5 foot cord. Before you say "Aha, finally a joystick that compares to an arcade type", read on... The basic Atari joystick's stiff movement, unrelible trigger button and it's predudice against left handed people are well known. The Atari joystick is the VW of joysticks.... very basic, cheap and durable. Wico offers an alternative with two firing positions - one on the base and another on top of the stick. It allows only firing from one position at a time. It uses six leaf switches as do the arcade machines. Weighing more than the Atari joystick, it suffers from the same problems as most joysticks - it is awkward to hold for any period of time. Will we always be lacking a solid base to shoot from? The games that I compared the two sticks on were Shamus, Krazy Shoot-Out and Star Raiders. The Wico out performed the Atari joystick in all these games . In Star Raider, hyperspace didn't make me as "hype". It was smooth sailing but still I was uncomfortable with the Wico. The shape is awkward and my hand was starting to cramp when I used the top button for too long. Is it worth the higher scores? My opinion is that the Wico is an ok investment for now, but I want somthing more and, as I have used the Atari joystick for four years now , I'm ready but still waiting.

A Visit From St. BASIC
By Jon Gauger

'Twas the night before Christmas.
Throughout my computer room,
Not a creature was stirring -
(That line you'd assume).
The CRTs sat in a state of despair-
Just hoping St. BASIC soon would be there.
The teletypes slept - computers in bed
While visions of "Do-Loops" danced through their heads.

Me in my P.J.s, my wife in her gown
were rather upset 'cause the Atari was down.
When way down the hall there arose such a clatter
I sprang from my bed to see what was the matter.

In one nano-second (well.. maybe 'twas two)
I was back to my hardware, my prized CPU.
The moon on the screen of my new CRT
Reflected so bright, like day I could see!
When what to my wondering eyes did appear,
But a huge floppy disk and a man growing near!

"Flying carpet from East," I murmured and cried;
But no, Saint BASIC himself had arrived.
The huge disc was powered my micro's in a train,
And he whistled and shouted and called them by name:

Now Franklin, now Apple, Atari and MITS!
On Radio Shack, Heathkit (and of course) the VIC!
From the big IBMs to the hobbyist's SOL
Now dash away, dash away, dash away all!!!

As human beings that, when hit with current, fly
When putting fingers in outlets, mount to the sky,
So down to my room, that company flew
With some software and hardware (and St. BASIC, too).

And then in a twinkling I heard on the floor
The squeak of his shoes as he opened the door.
As I drew in my head and turned it aside,
My computerist's room he walked right inside.

He was dressed all in jeans from his head to his toes
(Where he got all that denim, nobody knows).
Machine on machine he had stuffed in his sack
And he looked like a peddler opening his pack.

He was rather small - a runt, if you please,
But his eyes were lit up like big LEDs.
His hair was unkempt, but I didn't mind that,
I just sat there drooling at that big hardware sack.
His face was quite thin - like the rest of his bod
And pardon the word, but he looked like a clod!

A listing of some length, had that sly rascal.
I'm not very sure, but I think it was PASCAL.
A program tape he held tight in his teeth
And it wrapped 'round his head like an Arabian Sheik.

He looked rather tired and needed a shave,
But all of these faults I quickly forgave.
A wink of his eye - A twist of his head
Convinced me right off I had nothing to dread.

He spoke not a word but went straight to his work.
Fixed all my equipment and then turned with a jerk.
And closing the lid of his tool chest behumped
He picked up his sack - out the window he jumped.

He hopped on his disc - then input one line:
For Q equals one to a hundred and nine.
Flashed a huge CRT with a mighty big byte:
"Merry Christmas to all, and to all a good night!!!"

AT A TURTLE'S PACE WITH ATARI PILOT
by Curt Springstead-JACG

Welcome back to the land of terrapins (that's a $20 word for turtle). Hopefully after last month's column you have found a willing student or have resigned yourself to the fate that computing can be a lonely hobby. This month we will explore the capabilities of PILOT through a simple game proposed in the LOGO Ideas column in the November 1982 Creative Computing magazine.

The game as first created was called shoot and was used to aquaint people with the nature of the turn and go commands of LOGO. The computer would draw a circle somewhere on the screen and then place the turtle, as the graphic pen is known in LOGO and PILOT, somewhere else on the screen and face the turtle at a random angle. The player was to give the turtle commands for turning and movement until the turtle was inside the circle. Our version follows that basic format with a few changes and enhancements. This program also illustrates the relative ease of developing a program in small modules for each required function and integrating them in one program to complete the job.

During the developement of this program I created eleven different modules each performing one function of this program and several vain attempts to solve problems encountered during the design.

The first segment of the program performs some houskeeping and customizing functions. We reset the turtle to the origin or center of the screen each time the game is started just so we know where we are. The calculations in lines 40 to 70 set the colors for the four pens to more interesting hues. Feel free to experiment with these values to get colors to your liking. The instruction @B followed by a location allows you to alter or read the value of a specific location in the memory of the ATARI.

The next step was to randomly place the circle on the screen. This was easily done by turning the turtle to face some random angle from Ø to 36Ø. The '?' is the random number generator code and the '\36Ø' tells the system to select a value from Ø to 359 instead of the normal range of -32768 to 32767. The offset from the center of the screen is a random value up to 29.

The circle itself posed an interesting problem. Drawing a circle is easy, just imagine yourself a turtle and walk in a circle. You will notice that you walk straight a little, turn a little, walk and so on. The graphic turtle works in the same way. Now we have a circle, but how will we know if the turtle moves inside? There is entirely too much math for my taste especially with only integer math available. Well ATARI to the rescue, it seems that a special variable %Z was invented to allow you to see what color is under the turtle location you are at. It would be easy to know if the turtle was in the circle if the cicle were colored in, just test %Z at the turtle's location and see what color is there. The easiest way to color the circle would be to use the FILL command. In order to do this the FILL must find some color other than the background color to the right of the pen or the entire screen will be colored in. Therefore we first draw the right half of the circle with the normal draw command and then use the fill command for the left half of the circle. Viola a colored circle somewhere on the screen. The only cautions are the TURNTO 9Ø in line 13Ø to make sure the turtle is facing correctly for the circle drawing to work. The randomizing in step 15Ø simply further varies the location of the circle by determining if the starting point is the top or bottom of the circle. Any instruction placed before the colon is used to qualify the execution of the statement following and must be true for the line to execute.

```
10 *LOOP
20 R:SET REVISED COLORS FOR PENS
30 GR:GOTO 0,0
40 C:@B710=218 [BLUE PEN
50 C:@B712=86 [ERASE PEN (BACKGROUND)
60 C:@B709=0 [PEN YELLOW
70 C:@B708=0 [PEN RED
80 GR:CLEAR
90 R:TURN TO A RANDOM ANGLE AND MOVE UP TO 30
UNITS
100 R: TO LOCATE =TART POINT OF CIRCLE
110 GR:TURNTO ?\360
120 GR:GO ?\30
130 GR:TURNTO 90 [RESET FACING OF PEN
140 GR:PEN RED
150 C:#R=?\2 [HOLE TOP OR BOTTOM
160 GR(#R=0):45(DRAW1;TURN 4)
170 GR(#R=0):45(FILL 1;TURN 4)
180 GR(#R=1):45(DRAW 1;TURN -4)
190 GR(#R=1):45(FILL 1;TURN -4)
```

Now that we have the target circle drawn it's time to place the turtle on the screen. The same idea as the circle is used here, pick a random location and facing and draw a turtle. The only difference is the technique for locating the starting point. The screen coordinates run from -79 to 79 across the screen and from 47 to -31 from top of screen to the top of the text window. By taking a random value of 0 to 130 and subtracting 65, as in line 210, we will get numbers from -65 to 65 which will keep the turtle on the screen. Once the location has been determined the turtle can be drawn. A subroutine has been created to do this since we will need to draw the turtle many times in this program. Line 260 says use the routine at label *TURTLE and return here. Finally we are ready for the prompting of the player for his instructions.

```
200 R:GET RANDOM START LOCATION FOR TURTLE
210 C:#X=?\130-65
220 C:#Y=?\50-25
230 GR:GOTO #X,#Y
240 GR:TURNTO ?\360
250 GR:PEN BLUE
260 U:*TURTLE
```

We simply ask the player to enter a command consisting of the word TURN or Go and a number of units that command is to be executed through. The players command is placed by the ACCEPT command in the variable $INSTRUCTION for processing by the program.

```
270 *PROMPT
280 T:ENTER MOVEMENT COMMAND PLEASE
290 T:ENTER "TURN" OR "GO" WITH VALUE TO MOVE
EXAMPLE:TURN 45 or GO 20
300 A:$INSTRUCTION [GET INSTRUCTION
```

Line 310 looks for the space between the word and the value given. The system will normally return the first blank encountered if the underscore alone is used. The problem is that there is a "space" in front of the string as far as the system is concerned. To get to the second space or the first "real" space type ESC and a right arrow before the underscore. The MS: or match string command returns three pieces of information, $LEFT the data to the left of the match string, $MID the match string itself and $RIGHT the data to the right of the match string. In line 320 we accept the $LEFT value into $COMMAND and compare it in 330 to the list of actions we are looking for. Based on what item it matches the program jumps to the label assigned that position in the list. If $COMMAND contains GO then the program will jump to *GO. The JUMP command is like GOTO in BASIC the program moves to that instruction but does not return as with the USE command. The *PROMPT jump makes sure that all non-answers get handled properly, that is ignored.

```
310 MS: _
320 A:$COMMAND=$LEFT
330 M:TURN,GO,,
340 JM:*TURN,*GO,*PROMPT
350 J:*PROMPT
```

The drawing of the turtle is relatively simple. Draw a small circle and then another smaller circle on the front of the turtle and return the pen to the center of the body so that we can sense the middle of the turtle.

```
360 R:DRAW OR ERASE TURTLE BASED ON PEN COLOR
370 *TURTLE
380 GR:GO 4;TURN -90;GO 1;TURN 180
390 GR:30(DRAW 1;TURN 12)
400 GR:GO 1;TURN 180
410 GR:10(DRAW 1;TURN 36)
420 GR:TURN 90;GO -4
430 E:
```

Turning the turtle requires that the existing drawing be erased and a new one draw with the new facing. The pen is set to the background color, PEN ERASE, and the turtle is drawn, or is it undrawn? Anyway it's gone. The value for the number of degrees to turn the turtle exists in an alphanumeric or string variable and must be placed in a numeric variable before the program can use it. The calculations do not have a facility for placing the value of a string into a number so we must ACCEPT the value into the #A variable. The pen is set back to the blue pen color and the pen is turned the number of degrees requested and the *TURTLE routine is called to draw the turtle again. With the turtle drawn we can return to the prompt screen for the next instruction.

```
440 R:PROCESS TURN INSTRUCTIONS
450 *TURN
460 GR:PEN ERASE
470 U:*TURTLE
480 A:#A=$RIGHT [PLACE AMOUNT IN NUMERIC VARIABLE
490 GR:PEN BLUE
500 GR:TURN #A
510 U:*TURTLE
520 J:*PROMPT
530 E:
```

The go instruction moves the turtle forward the number of units specified by the command in the same way the turn commmand operated. The turtle is erased from it's current location and redrawn at the new location. The go instruction must also test to see if the new location places the turtle in the circle. To do that we must look at the color of the screen below the pen in the middle of the turtle. The system is instructed to lift the pen so that the screen is not redrawn to the pen color before we can look at the screen. The turtle is then instructed to GO 1 unit ahead of its last location and see what pen color it finds. %Z contains a value 0 to 3 for the pens erase, red, yellow and blue respectively. Since the circle was drawn with the red pen we are looking for a value of 1. When the turtle is found to be in the circle line 640 executes and control is turned over to the *HIT routine. If any other value is found the pen is returned to the center of the turtle and faced forward again in line 650 before returning for the next prompt.

```
540 R:PROCESS GO INSTRUCTIONS
550 *GO
560 GR:PEN ERASE
570 U:*TURTLE
580 A:#D=$RIGHT
590 GR:PEN BLUE
600 GR:GO #D
610 U:*TURTLE
```

```
62Ø GR:PEN UP
63Ø GR:GO 1
64Ø J(%Z=1):*HIT [TEST TO SEE IF IN CIRCLE,
PEN 1 AREA
65Ø GR:TURN 18Ø;GO 1;TURN 18Ø
66Ø J:*PROMPT
67Ø E:
```

When the turtle is found in the circle this routine prints a message and flashes the screen by alternately reversing the colors of the background and the circle. At the same time an alternating tone is produced. This fanfare continues for 7 cycles before the game resets for a new session.

```
68Ø R:HIT PROCESSING ROUTINE
69Ø *HIT
7ØØ T: A WINNER !!!!!
71Ø C:#I=Ø
72Ø *HITLOOP
73Ø R:SWAP COLORS IN PEN ERASE AND RED, VARY
TONES FOR WINNER
74Ø C:@B7Ø8=86
75Ø C:@B712=Ø
76Ø C:#A=1Ø
77Ø C:#B=#A+2
78Ø C:#C=#B+2
79Ø SO:#A,#B,#C
8ØØ PA:1Ø
81Ø C:@B7Ø8=Ø
82Ø C:@B712=86
83Ø C:#A=2Ø
84Ø C:#B=#A+2
85Ø C:#C=#B+2
86Ø SO:#A,#B,#C
87Ø PA:1Ø
88Ø C:#I=#I+1
89Ø J(#I<7):*HITLOOP [DO THIS 7 TIMES AND THEN
QUIT
9ØØ SO:Ø,Ø,Ø
91Ø J:*LOOP
92Ø E:
```

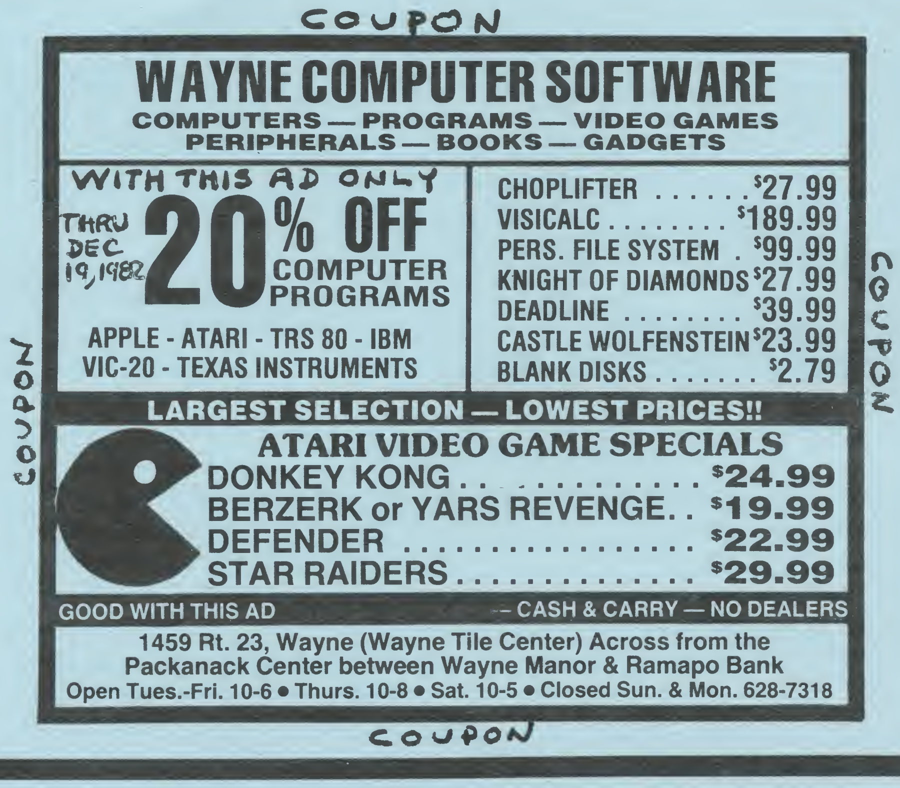
The program presented is very simple in concept and execution but seems to cover a multitude of functions within the PILOT language. You are encouraged to build on to the game by adding scoring in the form of move counts or timers to see who does the correction the fastest. The winner fanfare can certainly benfit from a little care. Next month I hope to report on alternate methods for entering the instructions with joysticks and paddles.

Well that's it for this month please let me know of your improvements to this program so we can all share and learn from your experiments.

1982-1983 MEETING SCHEDULE

All meetings are held on the second Saturday of each month in the Bell Labs Auditorium, Murray Hill, New Jersey. The meetings begin promptly at 1Ø:ØØ a.m. and end around noon.

Dec. 11, 1982
Jan. 8, 1983
Feb. 12, 1982
March 12, 1983
April 9, 1983
May 14, 1983
June 11, 1983
July 9, 1983
Aug. 13, 1983



ATARI MICROSOFT BASIC

Reviewed by Sheldon Leemon
Reprinted from M.A.C.E. June 1982

One of the features of the Atari Home Computer Sytem which reviewers often comment upon is that its BASIC language was not written by Microsoft. This is significant because Microsoft has written the BASIC interpreter for the Apple, the PET, the TRS-80, and for so many other popular micro computers that their version has become the de facto standard in the industry. This lack of a standard BASIC may have given some potential purchasers pause when it came to considering the Atari. Now, however, Atari has changed this state of affairs by adding Microsoft BASIC to the growing list of languages available for use with their Home Computer System.

Just what this will mean to Atari owners remains to be seen. One result will be a somewhat greater software compatability between the Atari and other popular personal computers. The instruction manual that comes with Atari Microsoft BASIC (I'll refer to it hereafter as AMB) even has appendices with instructions for converting programs from the PET, Apple, and TRS-80 to the Atari version. But despite the impression of compatibility that such instructions convey, the fact remains that Atari differs significantly from other computers in its hardware features, most notably in its graphics and sound capabilities. Programs which were written for other computers will not be able to take advantage of the increased user-friendliness that a generous helping of Atari sound and graphics can lend to a program, without a substantial reworking of the code. So while the implementation of Microsoft BASIC on the Atari will probably make available many useful and instructive programs, a thorough knowledge of the workings of the Atari computers will still be necessary to adapt these programs to take full advantage of the Atari's unique qualities.

Of course, the potential buyer will want to know not only what new commercial software AMB will make available for the computer , but also what advantages this BASIC dialect offers to the programmer who wants to write custom applications. This review will therefore try to highlight some of the more important features of AMB, to help you to decide whether it suits your own personal needs.

The initial question to ask yourself is whether you have a disk drive and at least 40K RAM. If not, AMB is definitely not for you--yet. The only version currently available is disk-based, and takes up about 19.5K of memory. With DOS booted up, that leaves only about 21K free in a 48K system, less if the RS-232 handler in the 850 interface is also booted up. Also, the disk is copy-protected, so you will always have to boot up with your master copy, and you will not be able to make back-up copies. You will also probably not be able to use this version with any of the new disk drives that require a patched version of DOS, including the Axlon RAMDISK board. Another side-effect is that in order to go to DOS, a MEM.SAV file must always be written, further slowing down the process of changing environments. If you find these limitations discouraging, don't give up just yet. Atari is already working on the next revision, which will be packaged in one 16K ROM cartridge, with a boot-up file of add-ons for disk users. This will allow owners of even minimum configuration systems to use Microsoft, with a minimum of fuss.

Before discussing the specific commands offered by AMB, a few words about Microsoft in general are in order. A number of features in Microsoft BASIC are implemented quite differently than in Atari BASIC. Those who have some experience with Microsoft on other machines will be quite comfortable with this version. Because it is based on the full, extended Microsoft model, it most closely resembles TRS-80 Level II, with a few more features. Those who have programmed primarily with the Atari 8K version are likely to notice a number of Microsoft's characteristic quirks as soon as they start to enter program from the keyboard. In Microsoft, no abbreviations are allowed, except the "?" for PRINT. No syntax checking is performed at the time of line entry, so errors in entry will not be apparent until the program is running. Also, Microsoft is much more particular about spacing than the 8K version, and failing to separate statements from numbers will result in a syntax error. When you do get the error message, however, you will be pleased to find that it is in plain English. Many statements can only be executed in the deferred mode, not from the keyboard.

Microsoft lets you choose the precision of numeric variables. Integer variables (indicated by the % as last character in the name) can be chosen for speed, or double precision (indicated by a # at the end) can be used to allow greater accuracy in mathmatical calculations. For convenience, the DEF- commands (e.g. DEFINT, DEFDBL) allow you designate all variables starting with one letter as having the same precision. Math functions are implemented by the interpreter, not by the slow floating-point package in the OS ROM, which makes faster calculations possible. Also, logical operators such as AND and OR do true bitwise comparisons, as opposed to the Atari operators, which compare the truth of the whole expressions. Logical true equals -1, not 1.

The sytem commands that Microsoft offers for "housekeeping" are very convenient. AUTO allows automatic line numbering for program entry, and will warn if a number duplicates an existing line number. DEL allows you to delete whole blocks of numbered statements. LIST allows open-ended listing such as 500- to list statements with a higher line number than 500. LOAD will load programs saved in either SAVE or LIST format. RENUM offers complete renumbering of programs, including references in such statements as GOTO, GOSUB, etc. And commands such as LOCK, UNLOCK, NAME and KILL allow you to perform the DOS functions that require an XIO command in 8K BASIC. Additionally, TRON and TROFF allow the tracing of program execution by printing on the screen the line number of the statement currently being executed.

There are a number of significant additions to the general program statements offered by AMB. For example, the MOVE statement allows you to copy any number of bytes from one location in memory to another. This is helpful in animating Player-Missile graphics, and in moving the ROM character set into RAM, so that user-created fonts may be employed. The IF..THEN sequence now allows an ELSE clause which will execute when the IF clause is not true. OPTION BASE allows you to choose whether array subscripts will start with a 0 or a 1. The WAIT command allows you to pause the program until a location in memory takes on a spe=ific value. This is ideal for halting execution until VBLANK occurs, so that graphics changes can be made without disrupting the display. It could also be used in conjunction with the real-time clock for a measured pause. There is another time-related statement which is even more fascinating. This is the AFTER command, which allows you to change the flow of program control after a given period of time. For example, the statement AFTER (600) GOTO 200 would have no immediate effect on the program. But in 10 seconds (600 jiffies, or 60ths of a second), the program would stop doing whatever it was doing and start executing at line 200! This gives you in effect a time-driven interrupt.

A rather remarkable addition in AMB is the COMMON statement. This allows you to designate certain variables which are to keep their values from one program run to the next. This greatly facilitates chaining several related programs together to operate off of one menu program, and will be most helpful in overcoming the RAM limitations imposed by the size of the interpreter. Other new program statements include ON ERROR, a slightly different error trapping scheme than TRAP, which allows you to RESUME the program after an error at the line where the error occured, the next line, or at any designated line number. ERL will return the line number where an error occured, while ERR will let you generate any error, for purposes of debugging your program's error trapping code. OPTION PLM, OPTION CHR, and OPTION RESERVE statements let you set aside reserved areas of memory for player-missilie graphics, character sets, or machine code.

I/O operations have been substantially overhauled in this BASIC, particularly as they relate to Atari's unique system for handling I/O in a device-independant manner through the use of a Central I/O Utility. For example, the OPEN command used to assign an I/O channel to a device now has a more English-like syntax, e.g. OPEN #1, "K:" INPUT. However, there is a price to pay for this simplification. Because the command syntax no longer corresponds directly to the requirements of the Central I/O Utility, you can only OPEN a device for the READ, WRITE, UPDATE, and APPEND functions, and not to read the Directory. You cannot OPEN the cassette for reading or writing files with short inter-record gaps. This would not be so bad if the XIO command had been retained. But because most of its functions have been implemented through other commands, it has been deleted. The only concession made to the BASIC user wishing to perform missing CIO functions such as formatting a disk, reading the disk directo=y, or reading or writing a block of memory to or from a disk file, is the inclusion of a disk file called CIOUSR. This file provides three prewritten USR routines which will allow you to condition the I/O Control Blocks and then call the Central I/O Utility. However, the user who is sophisticated enough to take advantage of the CIOUSR file is probably sophisticated enough to POKE the right values directly into the IOCB, and call CIO with his or her own USR statement. The upshot is that some I/O flexibility has been sacrificed for the sake of ease of use.

In other areas, I/O flexibility has been greatly enhanced. The INPUT command will let you substitute your own prompt message for the default question mark, and LINE INPUT will let you input a complete line, ignoring commas, quotation marks, and other terminators. There are a number of handy statements to aid in producing neatly formatted output. SPC will print a designated number of spaces. TAB will let you tab to a given print column. PRINT USING is fully implemented, letting you right justify, insert decimal places, trailing spaces, leading spaces, commas, dollar signs, asterisks, and generally line everything up in nice, neat, uniform columns. For screen output, the POSITION command has been replaced with PRINT AT(X,Y), where the variables X and Y indicate the screen row and column. Interestingly enough, this command also replaces the statement POINT, when used with a disk file as the device--in that case, the variables x and y indicate the sector and byte to write to.

The function library in AMB is mostly the same as 8K BASIC, although the math routines will generally have greater accuracy, and better speed. Trig functions are only available in radians in AMB. TAN is implemented. RND is a little more flexible, allowing you to generate random integers, and repeatable pseudorandom sequences as well. TIME is added as a function which returns elapsed time to 1/60 of a second. And the USR command is somewhat different in AMB. Rather than passing arguments to the machine language routine on the stack, AMB passes only one integer argument, directly onto two zero-page locations. While a programmer can use RAM from $CD to $FF in a USR call, there are no "safe" user zero-page locations, where variables can be stored immune to meddling from BASIC. As a matter of fact, not even page 6 is sacred--BASIC uses half of it, leaving little protected space for non-relocatable code.

One of AMB's great advantages over 8K BASIC is its ability to accept user-defined functions. This lets the user in effect make up his or her own BASIC commands. For example, as mentioned below, there is no AMB statement comparable to STICK(x), which returns the value of the joystick. However, the user could define such a function with the line DEF STICK(X)=PEEK(632+X). Then, every time thereafter in a program STICK(X) was used, the function would return the value of joystick number X. Moreover, the user could even define HSTICK(X) and VSTICK(X) as in BASIC A+, where the functions would return a 1, 0, or -1 depending on whether the joystick is being pushed up or down, right or left. One of the more serious oversights in the manual is the glossing over of this command. The functions which it allows you to define not only save time over a subroutine call, but can make possible the use of program constructs which otherwise could not be used. The DEF statment also allows you to define string functions, which will perform any allowable combination of string manipulations.

Perhaps the most significant difference between Microsoft and 8K BASIC is the way in which they handle strings. In 8K BASIC, strings are one dimensional, they must be DIMensioned, and can be as long as memory allows. In AMB, one-dimensional strings don't have to be DIMensioned, and true string arrays are implemented. This latter feature is greatly prized by Microsoft enthusiasts, who find that string arrays much simplify the task or character data manipulation. Maximum string length is limited to 255 characters, however.

In Microsoft, you don't have to allocate string space with a DIM command. Storage space is dynamically allocated by the interpreter. If the amount of free memory consequently gets low, AMB performs what is known as string-gathering or garbage collection, by which it reclaims sting space that had been previously allocated but which is now unused, and compresses the strings down to the space actually occupied by data. The programmer should be aware of this feature, as it may cause a noticeable pause in a program while it occurs. As an offshoot of this process, strings tend to move around in memory while a program is running, making them a less dependable place in whihc to hide machine-code subroutines, player-missile graphics, etc.

In addition to string arrays, many other handy string features are implemented. There is a true concatenation operator (C$=A$+B$). LEFT$, MID$, and RIGHT$ help separate out substrings, and INSTR performs a search for a substring within a larger string. STRING$ allows you to fill a string with any number of repetitions of a single character, or string of characters. Other functions now included within string operations are INKEY$, which records a keypress on the fly, TIME$ which returns the time in Hours : Minutes :Seconds format, and SCRN$, which replaces LOCATE, returning the value of data under the graphics cursor.

In the area of graphics, AMB is very similar to 8K BASIC. Major changes include the combination of PLOT and DRAWTO into a PLOT (X,Y) TO (X,Y) command that can be chained indefinitely within a single statement--a much streamlined procedure over the 8K BASIC version. A CLS command has been included to clear the screen. Also, SETCOLOR has been expanded to include the registers for player-missile graphics.

Player-missile graphics are supported to a moderate extent by AMB. The OPTION (PLM) commands set aside space for either single-resolution or double-resolution PM graphics, and puts the location of the base address into the proper hardware register. This address can be found with the VARPTR command (AMB's answer to the ADR function), and can be used to calculate the offsets into player and missile storage areas. As stated above, the SETCOLOR command can be used to control player color. The MOVE command can be used to shift the offset position of a player, to achieve vertical movement. But player width, priority control, horizontal position, collision detection, and other features still must be implemented through PEEKs and POKEs. Here is another area where the DEFined function could give the user a lot of aid. The manual thoroughly documents the features of PM graphics, and gives plenty of step-by-step examples for the beginner. This is a big improvement over the 8K BASIC manual, which does not acknowledge the existence of PM graphics. However, this manual still ignores the existence of the GTIA chips, and graphics modes 9, 10, and 11 which it makes available to users of machines made after 1981.

Last but not least we come to game controllers and sound. As mentioned above, there are no commands for joystick, paddle, or light pen reading, although they can easily be implemented with DEFined functions, or PEEKs. There has been, however, a pleasant addition to the SOUND statement--a fifth parameter that designates the duration of the

Continued on Page 14

THANK YOU!

JACG

FOR YOUR CONTINUING SUPPORT

JOYFUL HOLIDAY WISHES
and
HAVE A GOOD NEW YEAR

from:

Earthrise
Computer
Center

Phil D'Ambola, Pres.

FTWARE

ATARI®
AUTHORIZED
DEALER

DAY
IAL!

MINIMUM

25% off

MFR SUGGESTED RESALE

OLIDAYS

BOB & SARAH

note in 60ths of a second. This means that you no longer have to write imprecise delay loops to sustain sound effects.

The AMB manual is somewhat sketchy on the features of AMB that are common to all Microsoft BASICs, so that some prior knowledge of those features would really help the user. Such information is readily available from any of the great many books devoted to BASICs such as the one used by the TRS-80. On the other hand, as far as Atari specific features go, the manual is most detailed, giving explanations and examples concerning PM graphics, alternate character sets, a memory map, and information for translating programs from other BASICS.

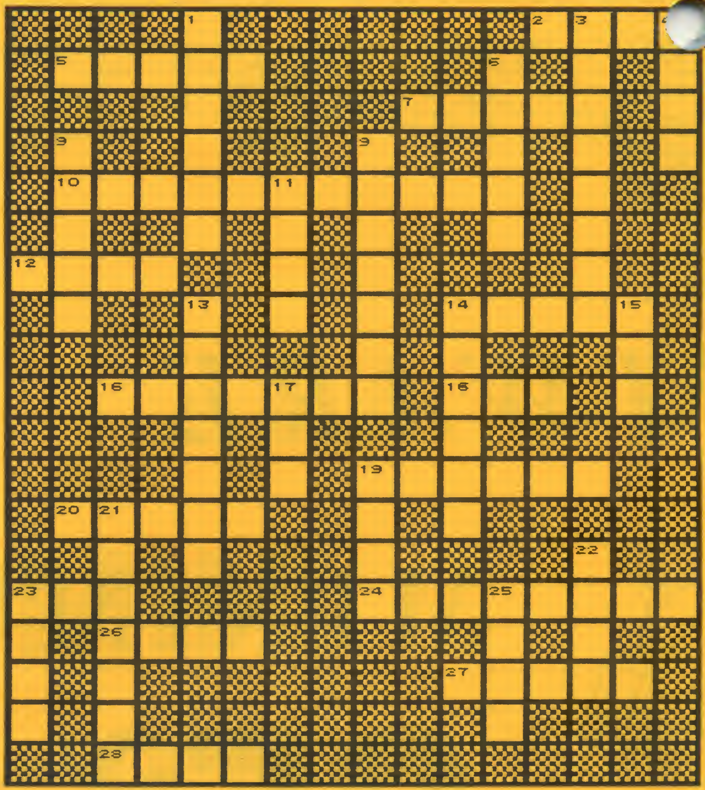
So there you have it. Atari Microsoft BASIC is large and feature-laden. Many users will find that the new commands far offset the trade-offs such as large memory requirements, limited string length, etc. For other users, who would like an extended BASIC, but who prefer the format of Atari 8k BASIC, BASIC A+ will probably be a better choice.

CROSSWORD MAGIC

from L & S Computerware - $49.95
Requires 40K, a disk drive and printer
Reviewed by Dick Kushner

I have just finished having a ball trying out a new program called CROSSWORD MAGIC. What this fantastic program does it allow you to create crossword puzzles in the comfort and privacy of your home. It is conveniently menu driven and allows you to do the following things with a puzzle: create, print, transfer, delete, complete, edit or play. The create-a-puzzle feature is truly amazing. Your start out with a 3x3 array and enter the words you want in the puzzle. It grows as you add words, right before your very eyes! If you don't like where the last word you entered was placed you can tell the program to reposition that word or delete it. If the word can't be fit into the existing puzzle, it is stored in memory and fitted in when there is room for it. You can, if you choose, have a fixed size of puzzle. After you have made the puzzle, you are give the opportunity to make clues for the words. All of this can then be stored on disk and/or printed out. The printing is another terrific feature. You can print out on any of about 20 different printers, all with graphics printing. Included are just about every printer you can think of except the Atari 825 (which doesn't have graphics printing capabilities). The printing is very sharp, as can be seen in the accompanying puzzle that highlights both the program and the Jersey Atari Computer Group. You can also save partially finished puzzles and go back and complete them at later time.

All in all, this program rates an A+ for concept, implementation and ease of use.



DOWN CLUES

1. HE HANDLES OUR ADVERTISING
3. HE IS OUR SECRETARY
4. TO DETERMINE THE CURRENT POSITION OF THE DISK FILE POINTER
6. WORLD'S BEST COMPUTER!
8. HE MANAGES OUR MONEY
9. EPSON OR CENTRONICS OR C-ITOH
11. DATA TRANSMISSION RATE FOR MODEMS
13. OUR GLORIOUS LEADER
14. THE NUMBER SYSTEM THAT ALL COMPUTERS THINK IN
15. WITH $, RETURNS THE STRING VALUE OF AN ASCII CODE
17. RETURNS THE LENGTH OF A STRING
19. OUR CALL LETTERS!
21. OUR NEW PROGRAM CHAIRMAN
22. DISPLAYS ALL OR PART OF THE PROGRAM CURRENTLY IN MEMORY
23. FASTER THAN A RECORDER
25. TO DISPLAY A POINT AT A SPECIFIED SCREEN LOCATION

ACROSS CLUES

2. ASSIGN AN I/O CHANNEL TO A DEVICE
5. SPECIAL ATARI SOUND CHIP
7. VIDEO DISPLAY CHIP
10. OUR NEWSLETTER EDITOR
12. OPPOSITE OF 26 ACROSS
14. BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE
16. OUR VICE-PRESIDENT
18. COMMAND TO ERASE CURRENT PROGRAM
19. THE "J" IN OUR NAME
20. OUR CHIEF LIBRARIAN
23. ABBREV. FOR OP. SYS.
24. WHAT THE ATARI EXCELLS AT!
26. IT RETURNS THE CONTENTS OF A MEMORY LOCATION
27. ESOTERIC, BUT VERY FAST ATARI LANGUAGE
28. USED BY A READ STATEMENT

The Choppers of Mercy

By Jim Salmons

Reprinted by permission from Softline Magazine

"War is a matter of statistics to those who give the orders and a matter of life or death to those who carry them out. Success of a mission is relative. Generals speak of 'minimal acceptable levels of casualties' to attain a military objective. Success is measured in terms of loss of life and equipment in pursuit of goals and ideals."

Fellini, Huston, Spielberg, move over: Dan Gorlin has arrived. Just when you thought it was safe to ignore the incessant stream of variations on the *Space Invader* theme, Gorlin decides to unleash his creative talents on the Apple to present us with what may well be the first Interactive Computer-Assisted Animated Movie. A fusion of arcade gaming, simulation, and filmic visual aesthetics, *Choplifter* is destined to occupy a place in the software Hall of Fame.

"Like most of us, I've watched a lot of television and movies, so I know what I like to see," Gorlin remarked recently. "And I just wasn't seeing what I liked in arcade-type games."

What Gorlin wasn't seeing was the use of the broad range of image processing preferences we have developed in this culture. These image preferences are like the implicit linguistic preferences we subconsciously apply when deciding whether an utterance sounds grammatically correct. Linguistic theorists speak of "performance" and "competence"—that puzzling ability to perform or speak a natural language without being able to verbalize the semantic rules that explain why certain words go together while others make gibberish.

Celluloid Dreamers. Having been exposed to a lot of television and film, most people in this culture have developed a "cinematic vocabulary," a set of predispositions to feel more comfortable when stories are told using certain visual images. In recent years, psychiatrists even report an increased frequency of patients describing dreams with detail befitting the shooting script for a movie, replete with close-ups, camera pans, and fade-outs. An intimate revelation is communicated between whispering lips in a close-up; a hand-held camera shot intensifies the frantic feel of a rowdy crowd scene.

A director can make or break a movie depending on how well his visual images enhance viewers' vicarious participation in the unfolding story. In this sense, Dan Gorlin might seem more like *Choplifter's* director than its author. He applied his skills as a programmer to his goal: to provide an interactive, animated story in which he would see what he liked on the video display of the Apple he had borrowed from his gadget-collecting grandfather.

Gorlin did not explicitly set out to make an animated, movielike game.

"I started fooling around with animated graphics on the Apple. I was intrigued by the challenge of doing a joystick-controlled helicopter. The physics of its motion is so complex and interesting.

"After I got the chopper flying around I had to put it somewhere, so I added the moving scenery. Then I added the tanks and planes. The idea of the hostages came last and really rounded out the scenario.

"I just kept adding what made it feel right to me. A story developed and I kept fooling around with the graphics until they looked right. Because the game developed a storylike scenario, movie camera techniques seemed appropriate, so the graphics grew out of the development of a scenario that required a movielike presentation. For instance, it just seemed right to have the final message be 'The End' instead of 'Game Over.'"

And just what might one expect from an Interactive Computer-Assisted Animated Movie game? Prior to purchase, magazine advertising and point-of-purchase packaging communicate an idea of what to expect from a new game. Too often, these sources of information are marketers' and artists' conceptions of what the game ought to be like to play. Too often, expectations are not met.

The Universal Hero. Unrealized potential is pleasantly not the case with *Choplifter*. The packaging is simple but effective. A wrap-around card protects the program disk, a full-color illustration depicts an action-packed battle scene on the front, and the back carries an enticing introduction urging the prospective buyer to become a hero—a hero of grand proportions.

The Bungeling Empire, a nation of militaristic ideologues, has kidnapped the sixty-four delegates to the United Nations Conference on Peace and Child Rearing. The hostages are imprisoned in four barracks. By virtue of an old treaty, the United States has a free-zone Postal Service building with protected status inside the territory. Three helicopters have been smuggled in disguised as mail-sorting machines.

The mission: fly into the empire, rescue the hostages, and return them to the post office so that discussions on world peace can begin in earnest.

Who can resist? Nasties holding nice folks whose "crime" is being interested in peace and kids. Throw in a slice of apple pie and just about anybody would jump into that chopper in the name of Truth, Justice, and the American Way. But remember, no one ever said being a hero would be easy.

The post office helipad and fluttering stars and stripes recede as the chopper crosses a pylon border that marks the Bungeling Empire. Twinkling stars and a full moon illuminate the background, while the scenery passes below.

The first chopper is not harassed by air attack, though Bungeling tanks can destroy it while it's on the ground to pick up hostages. Only after a copter has returned some hostages to safety do the Bungelings call in air support—streaking jet fighters that fire twin air-to-air missiles. With stray missiles and cannon shots exploding everywhere, the situation becomes a nightmare for the hostages.

The Sheep Look Up. The hostages are detailed little figures who behave amazingly like you would expect innocent folks to while running about in a scene where all hell is breaking loose. While the chopper is high and moving in, they pause to stand tall and wave as if to say, "Hey, we're over here. Please help us!"

These innocent victims then begin a desperate dash to meet the chopper wherever it lands. If the chopper takes off or a heated battle begins, some hostages may retreat, while others make a beeline in a last ditch effort to jump aboard despite tank blasts and air strafing.

When hostages make it to the chopper, they actually bend over and climb in the door. When the chopper has not landed solidly, a frantic group of hostages wildly scrambles at the chopper doors. If the pilot attempts to settle in such a hysterical crowd, lives can be lost under the landing gear.

Upon safe arrival back at the landing pad, the delegates pile out of the chopper. Most scamper straight for the front door of the post office; a few stop to turn and wave thanks to the hero for a job well done.

With just sixteen seats in the chopper, it takes at least four runs into enemy territory to rescue the hostages. If the tanks and jet fighters are unsuccessful in stopping the mission, Bungeling leaders unleash their most deadly weapons—drone air mines that float relentlessly toward the chopper, their presence creating deadly obstacles in crowded airspace until they're blasted away by the chopper's Gatling gun. Gatling gun?

That's right. The scenario calls for a hero, but even the peaceloving United Nations wouldn't send one in without some defensive weaponry. The rapid-fire cannon can be turned against the tanks and planes. And, like any real gun, it doesn't make value judgments. As its wielder, you must exercise extreme care not to produce "friendly fire." The mission is to rescue, not to kill.

Wings of the Dove. About Broderbund's role in the game's design, Gorlin explained, "They taught me about playability. They helped me with control of the joystick.

"The first *Choplifter* I showed Broderbund was too realistic, too much a helicopter simulation. De-emphasizing the weight of the calculations that simulated the vertical force control of the rotors made the chopper more flyable to the average player. I hated to see the realism go, but it did improve the game. In a lot of ways, Broderbund helped me fine-tune and polish the presentation."

And a polished presentation it is, the result of six months' program development.

In the process, Gorlin has exploited the potential of the two-button joystick; while most games provide joystick control as an option, *Choplifter* requires it.

Most computer games use a rather unreal movement, which, incidentally, signals the brain, "This must be a video game. I can't expect the ship to move the way it would in reality." Not so in Gorlin's creation.

The *Choplifter* joystick controls a helicopter, not a caricature of a helicopter on a video monitor. There's a distinction, and you can feel it the first flight out.

The chopper has a hovering feel to it. A sustained push of one button kicks the chopper around to face left or right, though the joystick allows left or right movement regardless of the chopper's orientation. This orientation is useful for covering the long distances between the post office and the hostage barracks. It's also the orientation used to counterattack the jet fighters that swoop out of the background and attack the chopper from the left or right.

A short button push puts the chopper into tank attack posture, facing front to the camera's-eye view. This makes it possible for the chopper to attack tanks in the foreground. With practice, you begin to develop an intuitive feel for the flight characteristics of the chopper, and the battle aerobatics become awesome.

Such subtle flight control is vital to a successful mission, because that Gatling gun, activated by the other button, is locked in the straight-ahead position. Aiming the cannon is accomplished by pointing the chopper in the appropriate direction and tilting its nose, bringing the gun into alignment with the target.

Of course, each aim and fire movement results in a follow-through motion. So the hero is caught in a harrowing game of cat and mouse where timing is everything. In the midst of this melee, you (you're the hero, remember?) must pick up the hostages without endangering their lives.

As a simulation, *Choplifter* forces player decisions, with resulting insights, well beyond those required in typical video games. There are situations in the heat of battle where split-second decisions must be made, such as whether to wait for the sixteenth hostage to scramble across the screen even as a tank and a couple of jets are fast approaching firing range. Risk all for one? Is being a hero dying for that last person or making it back with fifteen?

Such questions lead to insights not unlike the gruesome lessons to be learned from today's world news.

War is a matter of statistics to those who give the orders and a matter of life or death to those who have to carry them out. Success of a mission is relative. Generals speak of "minimal acceptable levels of casualties" to attain a military objective. Success is measured in terms of loss of life and equipment in pursuit of goals and ideals.

In the *Choplifter* scenario, the sixty-four delegates are worth the risk of three helicopters and pilots. If three choppers are shot down, the mission is scrapped, regardless of how many hostages are saved, killed, or still captive. If the helicopters survive, then as soon as hostages are safe or dead the scenario ends.

Scoring is simple. Only the number of hostages dead, the number in the chopper, and the number safe at the post office are displayed. You get no points for destroying tanks or planes.

Creator, programmer, director Dan Gorlin.

Your temperament and values determine whether aggressive behavior is warranted. Sometimes, you can't avoid it. On other occasions, it's righteous reflex, as in retaliation for an enemy tank having just obliterated a huddled mass of frightened hostages.

No matter what heroics were involved, when all hostages are accounted for or all choppers lost, a transformation occurs. The eyes of the hero turn into the eyes of the general reading the dead and rescued statistics. What is the measure of success? Were three helicopters lost worth the return of six hostages? Though sixty were returned, did four have to die?

The Answer, My Friend. The best resolution is sixty-four saved without losing a helicopter. The worst is total loss of choppers and hostages. In reality, the results seem inevitably in between.

So they seem in *Choplifter*.

Choplifter takes place in the midst of an altercation between hostiles. The situation requires you to fight. You are there because the enemy has committed an act of war. In all these things *Choplifter* is a war game.

Yet, in *Choplifter*, you join in battle to defend, not to destroy. Your goal is to save lives, not to kill. Your purpose is to prevent war, not to wage it. In all these things, *Choplifter* is an antiwar game—perhaps even a peace game.

If you look closely at the little people waving, you see their hope and excitement, their faith in you; you see their innocence. This makes *Choplifter* hard to play. It hurts to see one of those lively people killed by your landing gear or by the tank you missed. You may feel you want to turn the game off and go back to alien invaders; after all, you don't need to feel pain over a computer game.

But isn't this just a hint of how you'd feel if it were real? If you were the pilot and this were Iran two years ago? Shouldn't you go on anyway and get rid of the bad feeling not by tuning out, but by doing all you can to see the hostages safely home?

Maybe there is something we could do in real life, too, instead of turning off the seven o'clock news. ■

Ed Note:

Usually I don't print articles "wholesale" from other periodicals. However, I felt that this article was worth sharing with all JACG members.

Choplifter is a different kind of game from the usual "shoot-em-ups" that we all have come to know (and some of us, love). I had stopped playing video games on my computer, not because I think it is a sin or that playing them will cause me to go blind, but because I had lost interest in manipulating colored shapes that had no connection with reality. With Choplifter, the simulation of helicopter flight and the real-world depiction of an act of terrorism that has become much too common, combine to form a game that is, at once, both ego-involving and challenging.

In this game, your mission is to rescue as many of the 64 hostages as possible. You have 3 helicopters at your disposal but each return trip to the Post Office to deliver a load of hostages to freedom causes a higher level round to commence. Higher level rounds contain one or more enemy jet fighters, multiple enemy tanks and heat-seeking enemy air mines which, after the 4th round, not only fire at your chopper but appear with increasing frequency (and even follow you back to the Post Office). No points are earned for destroying the fighters, tanks and air mines - the only priority is saving those helpless hostages.

I would suggest that you reread this article and if you have not yet purchased the game go out and buy it. Then think about how you would combine the individual measures of performance (the number of hostages saved, killed and left behind, the number of helicopters used, and the player's technique- rescue only or aggressive) into one overall score. Let me know what you come up with.

Softline is a bimonthly magazine by Softalk Publishing which focuses on entertainment software for home computers. Subscriptions are $12.00 per year (6 issues) and can be obtained from: Softline Circulation, P.O. Box 60, North Hollywood, CA 91603. This article originally appeared in the July 1982 issue and I would like to thank Ms. Margot Comstock Sommervik for permission to reprint it.

After you get this game, I'll see you back at the U.S. Postal Service helipad.

==============
TRADING POST
==============

Trading Post is a monthly column that allows JACG members to list items for sale or swap. There is no charge for this service, and material must reach me by the 20th of the month.

For Sale: New Atari 825 80 column printer with interface cable and custom made static guard cover. $425.00 Call: Amato DiLauro, (212)626-7545 (after 6PM) or (201)468-3709 (until 4PM).

For Sale: Atari 400, BASIC cartridge, Joysticks, Missile Command or Star Raiders Game - all for $225.00. Contact: John Griffith, (201) 297-9349 (evenings).

Wanted: Donations of hardware/software for the Atari 400/800. Tax-deductible donations are for The Swain School, 1100 S. 24th St., Allentown, PA 18103. (215) 433-4542. Contact: J. D. McElroy - (201) 386-2865. Thank you.

ATARI BULLETIN BOARD SYSTEMS LIST
by Bob Reeves - JACG

Here's a list of Atari BBS's compiled from various sources. If you have any additions, deletions, etc., please leave a message for SYSOP on JACG BBS or call Bob at 201-689-0291.

Compiled on Aug. 30, 1982

Type	Name	Loc.	Phone
AMIS		GA	404-252-9438
AMIS		LA	504-273-3116
AMIS		IL	312-789-3116
AMIS	GRASS GD.RAP.	MI	616-241-1971
AMIS	MLBBS	WI	608-251-8538
AMIS	STRBASE 12	MA	617-876-4885-L
AMIS	TEAM	CA	408-942-6975-L
AMIS		WS	509-582-5217-L
AMIS		IL	312-789-3610
AMIS	GRAFEX	CA	408-253-5216
AMIS	APOGEE	FL	305-238-1231-RL
AMIS	ARCADE	MI	313-978-8087-RL
AMIS	MACE	MI	313-868-2064-*
AMIS	SPACE	WA	206-226-1117
AMIS	JACG	NJ	201-377-4084-L
ARMU	ARMUDIC #1	DC	202-276-8342-*
ARMU	ACE-NET	PA	412-655-2652
ARMU	COMPUTER AGE		301-587-2132-L
ARMU	ARMU PACE	PA	412-655-2652-*
ARMU	GREKELCOM	OK	405-722-5056
ARMU	FLEDGLING	NYC	212-598-0719-L
ARMU	SYNAPSE (AD)	CA	415-527-8276-L
ABBS			404-252-9438
ABBS		HI	808-833-2616
RCPM		IL	312-789-0499
RCPM		MI	313-759-6569-R
RCPM		PA	215-398-3937
TARI-BOARD		GA	404-252-9438
TARI-BOARD		CO	303-221-1779
TARI-BOARD		NJ	609-324-5875-L

Sorry, incomplete info on the following, but you might want to try them:

TYMCOR	MA	617-738-5051
MICROBITS	OR	503-967-9075
BEECHWOOD		216-582-2797
???		416-533-9216

L=LIMITED HOURS OF OPERATION
*=24 HOURS
R=RINGBACK. To contact:

1. Call listed number
2. Let ring one time
3. Hang up
4. Call back within 15 secs.

Again, please notify us if you have additions, deletions, changes, etc. so that we may keep this list accurate and up-to-date.

****
DON'T FORGET

J A C G BULLETIN BOARD

(201) 377-4084

Monday-Sunday: 9pm-9am
****

K-DOS by K-BYTE
Reviewed by Sheldon Leemon
Reprinted from M.A.C.E. July 1982

K-DOS makes an interesting follow-up to K-Byte's first release, K-razy Shootout. As you might have guessed, this is not a game, but rather an alternative to Atari's Disk Operating System, DOS II. While its file management system is compatible with Atari DOS, K-DOS offers a greater level of control over devices and memory, and is structured so that it is easy to use. Some of the features it offers will be appreciated by every Atari user, but many are of interest only to the serious programmer. Since the benefits to be gained by using K-DOS come at the price of certain trade-offs, the potential buyer should think hard about how much a more convenient disk operating system is really worth to him or her, before taking the plunge.

Chief among these trade-offs is the amount of memory it leaves available to the user. K-DOS is memory resident, and while that make most of its features immediately accessible, it also takes up a lot of space. With a BASIC cartridge inserted, the amount of free memory available in a 40K system is 25228 bytes. This is almost 7K less than the 32274 bytes available with Atari DOS, or the 31758 bytes available with OS/A+. Besides reducing the amount of memory available for programming, K-DOS' large size puts the start of low memory above $3000 (although an optional program included with the package will let you remove the plain English error messages, which saves enough bytes to bring the end of K-DOS just below $3000). Because many machine-language programs are assembled to run just above the end of Atari DOS, these programs may conflict with K-DOS, and may nct run under it.

If you have the memory to spare , however, KDOS has many attactive features to offer. The improvements in human engineering make it a pleasure to use. The best example is the fact that any of thg DOS functions are accessible from BASIC, PILOT, the Assembler cartridge, or whatever program environment you happen to be in. Since K-DOS is command driven, you don't have to call up a menu execute a DOS function. All you have to do precede the command by a comma (or some other character which you can define as significant to DOS), and the DOS function will be executed without changing program environments. The syntax required for command lines is flexible, so that commas can be replaced by a space, lower case is acceptable, and the DOS environment automatically resets the inverse character shift. Device defaults are supplied whenever possible, and short abbreviations are allowed, so that a minimum of keystrokes are required to perform any given function. Error messages come back in plain Enlish, rather than a frustrating number code. Unlike OS/A+, which puts you back in the DOS environment every time you hit System Reset, K-DOS only will bypass BASIC if you hit the Start key along with System Reset. And unlike Atari DOS, the device handler for the 850 interface unit will be booted automatically at power-up if it is turned on, without the need for a separate AUTORUN.SYS file.

The reason that K-DOS can let you use DOS command lines from BASIC is that reroutes all input to the line editor (although it gives you a command, KILL which will take its "hooks" out of the handler table if desired). This greater level of control over the system is characteristic of K-DOS. For example, the 6502 BREAK instruction is vectored to get you back to DOS any time the instuction is encountered, rather than have the system "hang up". You may get a little better idea of what this means if you slip in the BASIC cartridge and type INPUT (carriage RETURN). With Atari DOS II, the system locks up, and the only way to recover is to turn the computer off and reboot. With K-DOS, a BRK message is given, and you enter DOS. You should even be able to recover from the dreaded "editing lock-up", which occurs when BASIC moves a block of exactly 256 bytes (you will still have to know enough about how BASIC works to reset the statment pointers, however, as that particular bug tampers with your program code before it crashes the system). Another aspect of K-DOS' system control is that it allows you to stop disk I/O just by hitting the break key.

ithout destroying your data. It also tries harder to read and write marginal sectors before bombing out, which is important given the notorious speed fluctuation of the older Atari disk drives.

K-DOS lends nice new touches to some of the original DOS functions. For example, INIT combines formatting and writing DOS files to the new disk in one operation, although these functions are still available separately. The duplicate disk function allow you the option of a straight sector copy, for boot-disks that do not have file information on them, and also allows the faster write without verify, and continuous retrying of bad sectors. There is a separate APPEND command, and it allows you to enter data on to the end of a file directly from the keyboard. Also, the append function uses any space available in the last sector, rather than starting a new sector like Atari's does. The binary load command will print to the screen the location in memory into which the file is being loaded, if you so desire, which is a lot more convenient than reading the headers and calculating the addresses by urself.

But K-DOS doesn't take up all that memory just to give slight modifications. It also contains a complete machine language monitor. This allows you to examine memory in hexadecimal and ASCII formats, to alter memory by typing in either hex or ASCII values, and to examine and alter the contents of the registers as well. It give you two ways to execute a machine language program. GO runs the program after closing all devices, and does not preserve the registers. PROCEED will continue a program after a breakpoint has been reached, without changing the contents of the registers or the status of any device, making it a very handy debugging tool. Similarly, the command XIT will let you get back to a BASIC program that calls DOS, and will continue to run that program from the point at which DOS was called. A null device handler has been added, so that you can test I/O operations quickly by directing them to N:. LOMEM lets you examine and alter the bottom of memory available to a cartridge. This allows you to reserve space for achine-language programs, or just to reduce the amount of memory available to see if a BASIC program will run on the minimum 16K system. UDC allows you to add your own user-defined commands to the system.

In addition, K-DOS offers many commands which allow you to access certain routines used internally by DOS, just by giving a one-word command. For example, COLD and WARM provide an easy way to coldstart or warmstart a cartridge. RESET reboots the 850 handler when you have expanded the drive buffers, or just forgot to turn it on when you booted up. TEXT corresponds to a GRAPHICS 0 call in BASIC, and opens the screen device, which is handy for moving the display list when you want to load a program into high memory. CLOSE closes all files, turns off the sound, resets VBLANK vectors, and turns off Player-Missile graphics. ER followed by a number will print the Enlish error message for that error number, which is very handy when you want to interpret I/O errors that are generated by BASIC. None of these functions are earth-shaking, and all can be otherwise accomplished with a little effort, but the author's attitude was that as long as the routines for doing them were already in DOS, it made sense to allow them to be accessed easily.

Unfortunately, the lack of depth in the documenation runs somewhat counter to this intention of allowing the programmer easy access. The glossy K-DOS Handbook is nicely bound, comes with a pocket summary card, is clearly written, gives examples of the proper syntax for each command, and covers most of the commands very well. But it treats some of the more esoteric commands in a cursory manner. Take, for example, the handbook's explanation of the UNLOAD command: "Tries to erase area where cartridge is; unloads any RAM based cartridge and resets LOMem back to end of DOS." The beginner will no doubt read this sentance, re-read it once to verify that all of the words are in Enlish, and then press on, no better or worse for the experience. The sophisticated user, on the other hand, might gather from this explanation that it is possible to load a program into RAM, and fool the system into thinking that it is cartridge based, allowing an easy transition back and forth between that program environment and DOS. The inference would then be that the UNLOAD command erases this program, and lets the system know that no cartridge is present. But how do you set up this "RAM based cartridge" in the first place? No clue is given, leaving the sophisticated user perhaps more frustrated than the beginner. Another example in the same vein is the system equate files that are supposed to give the user access to system routines, such as the one to type text messages from a buffer. There are no detailed examples included of how to use them, however, and the internal commenting is too scanty to allow most users to benefit from them. Features like these could be real selling points to the ambitious programmer if treated less superficially in the documentation.

My subjective impression of K-DOS is that aside from these omissions in the documentation, it is a convenient tool for the user who is serious about programming. As one who uses his computer mostly for programming, I have found it especially helpful in developing software that combines BASIC with machine language subroutines. But I think that KDOS will be of much less interest to the casual programmer who doesn't have at least 40K of memory. While such a user might really appreciate some of the features, he or she would probably never take advantage of the machine language monitor, the null device, or many of the other goodies which make K-DOS so big, and so expensive. If you fall into that catagory, you would probably be more satisfied spending the money on something that would let you gobble dots, eradicate insects, or save the universe.

PROGRAM
DIRECTOR'S REPORT

Since this is a newly created post, it will take some time to formalize the procedures for each meeting. In the meantime I have several ideas on what I think the format should be. The president's report will include news from ATARI and any other pertinent information. This will be followed by the main topic or speaker for the month. This should leave 30-60 minutes for game and program demonstrations. There are several reasons why I feel this format will work, but the main reason is the number of people who leave when the meeting begins to run past 12:00. If a person has volunteered to speak to the group I feel that s/he should be given the time to cover the topic. If anyone has to leave, they can always preview new games at the local outlets.

The best format is one that serves the members needs. Therefore, I look forward to your suggestions on what you would like to see. I intend to publish the next month's meeting program so you can prepare questions ahead of time in order to get the most out of a talk.

We already have some interesting speakers scheduled for future meetings. Included are: a representative from the Atari Technical group in New York, a service representative from the service center in Somerset and a representative from Compuserve. The Compuserve Rep will explain and demo the services that the company offers which, based on information from the OHIO users group, includes the availability of group discounts and special considerations.

I will be near the podium at the beginning of each meeting, so feel free to give me your suggestions.

Richard Rospond
Program Director
(201) 635-2936

TITLE PAGE
by Gary Kunkel - JACG

After using the program by Frank Hopwood (JACG Newsletter V2N2P6) that creates an AUTORUN.SYS file, I decided to write a program that would show a title page and a menu of options. The options are: go to BASIC,go to DOS, and list the contents of the Disk Directory.

Type in this program, save it, then type in and run the AUTORUN.SYS file program which will give you directions on how to make an autorun file out of this TITLE PAGE program.

```
1 REM  Title Page
2 REM By Gary Kunkel
3 REM
5 REM Set up screen for mixed GRAPHICS modes of 2 and 0 on GRAPHICS 1 background.
6 REM Two rows of GRAPHICS 2 and fourteen rows of GRAPHICS 0
10 GRAPHICS 17
20 BEGIN=PEEK(560)+PEEK(561)*256+4
25 POKE BEGIN-1,71
30 POKE BEGIN+2,7:POKE BEGIN+3,7
   ROW=6 TO 20:POKE BEGIN+ROW,2:NEXT ROW
   E BEGIN+22,65:POKE BIGIN+23,PEEK(560):POKE BEGIN+24,PEEK(561)
55 REM background color is dark blue
60 SETCOLOR 4,9,1
62 REM Poke the number of the graphics mode in which you are to print into memory location 87
63 REM You may print whatever you like after the PRINT #6 command.
64 REM Just change the first parameter of the POSITION command to center it.
70 POKE 87,2:POSITION 5,0:PRINT #6;"ATARI 800":POSITION 1,1:PRINT #6;"personal computer"
75 POSITION 8,2:PRINT #6;"48K"
80 POKE 87,1:POSITION 4,3:PRINT #6;"gary kunkel":POSITION 4,4:PRINT #6;"DISKETTE #1"
83 REM BlankS out cursor.
85 POKE 752,1
87 REM Menu of options.
90 POKE 87,0:POKE 82,0
110 ? "Type in B for BASIC."
120 ? "Type in D for DOS."
130 ? "Type in L to list the Disk Directory."
140 ? "In order to temporaraily stop the"
150 ? "listing, hit [CTRL] and 1"
160 ? "at the same time."
195 REM Subroutine to input the menu  options.
200 OPEN #3,4,0,"K:"
220 GET #3,B
240 IF B=66 THEN GOTO 500
250 IF B=68 THEN DOS
255 IF B=76 AND A=1 THEN TRAP 1010
260 IF B=76 THEN GOTO 1000
265 IF B=84 THEN GOTO 2000
267 REM The numerical values in B are the ATASCII values for their corresponding letters.
270 GOTO 220
300 END
400 REM Option when "B" for BASIC is  typed.
500 GRAPHICS 0:POKE 82,2:NEW
700 REM List the contents of the Disk Directory.
1000 DIM A$(20)
1010 POKE 82,2:GRAPHICS 0
1020 OPEN #2,6,0,"D:*.*"
1030 TRAP 1065
1040 INPUT #2,A$
1050 PRINT A$
     A=1
     GOTO 1040
1065 CLOSE #2
1066 REM Options after listing the    contents of the Disk Directory.
1070 ? "Type in B for BASIC."
1071 ? "Type in D for DOS."
1072 ? "Type in L to list the Disk Directory."
1073 ? "Type in T for the Title Page."
1080 GOTO 220
1090 REM Option when "T" for Title    Page is typed.
2000 CLOSE #3:GOTO 10
```

Ed Note: Portions of this program were obtained from COMPUTE' magazine and the book - Your Atari Computer by Poole, et. al.

THIS SPACE INTENTIONALLY LEFT BLANK

A BLINKIN' CURSOR ROUTINE
by Frank Hopwood -- JACG

After sitting a good many hours in front of a commercial word processing system (WANG), I've come to appreciate the value of a blinking cursor in finding my place on a crowded screen.Especially after having looked away for a moment. As a matter of curiosity, more than of need, I undertook the challange of equipping a few menu-driven programs with this little enhancement. If nothing else, I reasoned, the added motion might help to liven-up my input routines a bit.

Below, is an annotated listing of the blinking cursor routine, for those of you who might want to add that little extra something to an otherwise ho-hum input statement.

```
1 REM *** BLINKING CURSOR ROUTINE ***
2 REM ***     By Frank Hopwood     ***
3 REM
10 DIM CHOICE$(1)
20 ? CHR$(125):REM CLEAR SCREEN
30 POSITION 2,11:? "END BLINKING CURSOR DEMO? (Y/N)";
40 GOSUB 1000:REM BLINK UNTIL KEY PRESSED
50 CHOICE$=CHR$(BYTE)
60 POSITION X+2,Y:? CHOICE$:REM PRINT CHOICE$ AT
CURSOR LOCATION
70 FOR DELAY=1 TO 100:NEXT DELAY:REM HOLD PRINTED
CHARACTER
80 IF CHOICE$<>"Y" THEN 20
90 POKE 752,0:REM RESTORE ORIGINAL CURSOR
100 END :REM GOTO MAIN PROGRAM FROM HERE
1000 POKE 752,1:REM TURN OFF CURSOR
1010 X=PEEK(91):Y=PEEK(90):REM LAST POSITION OF CURSOR
1020 OPEN #1,4,0,"K:"
1030 POSITION X+2,Y:REM OFFSET NEW CURSOR TWO SPACES
(OR MORE, e.g., X+n)
1040 ? CHR$(160);:REM PRINT NEW CURSOR
1050 FOR BLINK=1 TO 50:NEXT BLINK:REM HOLD CURSOR "ON"
1055 REM THE TRICKY PART
1060 IF PEEK(764)<>255 THEN GET #1,BYTE:POKE
764,255:CLOSE #1:RETURN
1070 POSITION X+2,Y:? " ";:REM PRINT SPACE IN PLACE
OF CURSOR
1080 FOR BLINK=1 TO 50:NEXT BLINK:REM HOLD CURSOR
"OFF"
1090 GOTO 1030:REM ONE BLINK COMPLETED; BEGIN ANOTHER
```

CONTEMPLATING A MODEM

by Bob Reeves - JACG

I'm writing this article for anyone who is contemplating the purchase of a modem for their computer. The first thing that I did was gather all my past issues of Compute, Analog, Antic, etc., and read over all articles on modems and telecommunications. What I found was a lot of general information, but not very much in the way of the more specific practical information that I really needed. I had pretty much decided that I was going to buy a modem, and was trying to decide what type to buy, and just what I would do with it on a day to day basis.

Based on what I had read, I decided on a direct connect rather than an acoustic. This modem connects directly to the telephone line and works independently of the telephone. The acoustic modem works through the mouth and earpieces of the telephone, and because of this, the reliability is said to be somewhat less. I know people who have acoustic modems, none of whom have had any problems, but the direct connect is still, in principle, more reliable.

The next choice is whether one needs auto-dialing and auto-answer features. I decided to opt for both, so that the modem could operate under computer control, as would be required to operate a bulletin board system or to answer calls from an office computer. I occasionally use a computer in my lab at work, and wanted to leave open the possibility of communicating between lab and home computer (and maybe even getting my company to pick up the tab for a dedicated phone line at my home). Well, maybe if the economy improves a bit.......

My final choice was the Hayes Smartmodem, and I would highly recommemd it to anyone who requires its features. It can communicate with almost any type of computer, with the proper program. Many bulletin board systems (BBSs) use this modem, which says something for it. There are however, many other makes of both direct and acoustic modems, none of which I have any personal experience with. I have seen ads in the magazines for a direct connect modem for less than $1ØØ (Signalman, I think). I don't think you can by an acoustic for that price!!

I should emphasize that you will need a terminal program of some kind to operate any modem. I started with TELELINK, which I think is an excellent choice for a beginner. It does not allow you to upload or download files but you can talk to the SOURCE or COMPUSERVE (an hour on each is included free with the purchase of TELELINK, and is almost worth the price of the cartridge), and you can communicate interactively with other users or BBSs. You will also need the ATARI 85Ø Interface Module.

Now on to the fun stuff!! What do I do with my modem? I'm glad you asked. I help keep the price of AT&T stock up. My phone bill the first month that I had a modem went from $5Ø to $15Ø, so be prepared. I figured it's the least I can do for a company who donates their entire auditorium to my computer club once a month!

I spend most of my modem time with Atari bulletin boards. These are computer run bulletin boards set up by hobbiests such as ourselves, and operated either out of someone's home, or from a gracious computer store owner's place of business. These bulletin boards generally offer message files, where users can exchange ideas, hot news, and the like, and a menu of programs that users can download (if they have the proper terminal program). There are several public domain terminal programs, JONESTERM.III being one of the most popular. I have personally used this one and it is excellent (and you can't beat the price). A good terminal program allows you to download files from or upload files to a BBS. Files are generally sent in the Basic LISTED version, so after transferring to disk or tape, they must be ENTERED rather than LOADED into memory. This forma also allows the user to edit the program in familia form in case a bad character or two comes across. All BBSs have specific instructions for uploading and downloading files, and it's a good idea to save the instructions to a printer for future reference, or have a friend print the instruction file for you if you don't have a printer.

I have downloaded many fun games and useful utilities from BBSs, all for the price of a phone call, which is really a buy if you look at the price of an average piece of software today! You won't find PACMAN or STAR RAIDERS, but there is some really good public domain stuff out there, and the best of it is on BBSs. Like Stan Ockers programs!! Just call the ACE BBS and download your choice! Need a good RPM program for your 81Ø drive? You'll find one on several BBSs.

In conclusion, modems are useful, informative, time-saving, and most of all, FUN!! You probably can't go wrong with any model that is presently being advertised. I have yet to read a bad review of one. Just make your choice on what features you now need and anticipate you might need in the future if you get hooked, and have fun!

POKING AROUND

By Arthur Leyenberger

-+-+-+-+-+-+-+-+-

This will be a regular monthly column that will feature interesting and useful PEEK and POKE locations. This month I will present just a few. (Is there a JACG member out there willing to write this column on a monthly basis?)

POKE is the instruction used to place a number int memory. Decimal numbers between Ø and 255 are use If your Atari gets confused, you can usually pres RESET and the original (default) values will be restored.

To set the left margin of your television or monitor screen use POKE 82,n-- where n is the column you would like to be the left margin. (Default value is 2).

To set the right margin, use POKE 83,n-- where n is the right margin column. (Default value is 39).

To set radians mode, use POKE 251,Ø. POKE 251,6 puts the Atari in degrees mode. (Default value is Ø, radians mode)

To turn the cursor off, use POKE 752,1. To turn the cursor on use POKE 752,Ø.

POKE 2Ø2,1 in your BASIC program will cause the entire memory to freak out if the user presses either BREAK or SYSTEM RESET.

To perform a system reset from your BASIC program, use POKE 58Ø,1. This will run any AUTORUN.SYS file you have on the disk, like the menu.

If you would like to double the rate at which data is written to the disk, POKE 1913,87 will eliminate the read-after-write verify on disk I/O. To make this change permanent on a disk, after doing the POKE, go to DOS and choose option H, WRITE DOS FILES. To restore the read-after-write verify, POKE 1913,8Ø. If an error occurs during disk I/O without verification, ERROR 145 will result.

That's it for this month. Next month we will offer some more exciting POKE locations and discuss PEEKS.

```
#####################################
#         J   A   C   G             #
#  JERSEY ATARI COMPUTER GROUP      #
#         58 DEWEY AVENUE           #
# HIGH BRIDGE, NEW JERSEY Ø8829     #
#####################################
```

FIRST CLASS MAIL

JACG NEWSLETTER—VOLUME 2—NUMBER 4
DECEMBER 1982

The Jersey Atari Computer Group (JACG) is an independent, informal organization of Atari computer users. It is not affiliated with Atari or any other commercial organization. Any logos, trademarks or company names are used either with permission or as an informal way of referring to a product or organization. Opinions expressed in this publication reflect only the views of the individual author, and do not necessarily represent the views of JACG.